

Amendment and Response under 37 C.F.R. 1.116

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

IN THE CLAIMS

Please amend claims 1-6, 8-11, 17-23, 25-31, and 33-34 as follows:

1. (Currently Amended) A method of managing memory in a distributed computer system, the method comprising:

binding a remote key to a first address representing a contiguous memory address range accessible by a first consumer process stored in a first memory at a first host processor endnode including a first processor and the first memory;

sending the bound remote key and first address from the first host processor endnode to a second host processor endnode on a communication fabric, wherein the second host processor endnode includes a second processor and a second memory; and

performing a remote direct memory access operation from the second host processor endnode with a second consumer process stored in the second memory ~~from the second endnode~~ to access the contiguous memory address range including sending the bound remote key and the first address from the second host processor endnode to the first host processor endnode on the communication fabric.

2. (Currently Amended) The method of claim 1 further comprising:

controlling local memory access protection in the first host processor endnode with a virtual memory manager in an operating system kernel process stored in the first memory.

3. (Currently Amended) The method of claim 1 further comprising:

comparing the bound remote key and the corresponding first address supplied by the second host processor endnode to the bound remote key and corresponding first address in the first host processor endnode.

4. (Currently Amended) The method of claim 3 wherein if the bound remote key and corresponding first address supplied by the second host processor endnode do not match the bound remote key and first address in the first host processor endnode, the second host processor endnode is not granted access to the contiguous memory address range.

Amendment and Response under 37 C.F.R. 1.116

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

5. (Currently Amended) The method of claim 3 wherein if the contiguous memory address range represented by the first address bound to the remote key supplied by the second host processor endnode is invalid, the second host processor endnode is not granted access to the contiguous memory address range.

6. (Currently Amended) The method of claim 1 wherein the first address is an effective address pointing to an address space in the first memory accessible by the first a consumer process.

7. (Previously Presented) The method of claim 6 wherein the effective address points to a virtual address space.

8. (Currently Amended) The method of claim 7 further comprising:
comparing the bound remote key and the corresponding first virtual address supplied by the second host processor endnode to the bound remote key and corresponding first virtual address in the first host processor endnode; and
handling a page fault condition in the first host processor endnode caused by the first virtual address bound to the remote key supplied by the second host processor endnode not being previously mapped by an operating system of the first host processor endnode.

9. (Currently Amended) The method of claim 16 wherein the first consumer process is a user process.

10. (Currently Amended) The method of claim 16 wherein the first consumer process is a kernel process.

11. (Currently Amended) The method of claim 1 wherein the first address is a virtual address accessible by the first consumer process which is a consumer kernel process.

Amendment and Response under 37 C.F.R. 1.116

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

12. (Previously Presented) The method of claim 1 wherein the binding includes associating the first address to the remote key with a consumer process employing a bind remote key verb.
13. (Previously Presented) The method of claim 1 further comprising:
obtaining at least one remote key with a consumer process employing an allocate remote key verb.
14. (Previously Presented) The method of claim 1 further comprising:
unbinding the remote key from the first address with a consumer process employing an unbind remote key verb.
15. (Previously Presented) The method of claim 13 further comprising:
retiring at least one remote key that was previously obtained via the allocate remote key verb with the consumer process employing a deallocate remote key verb.
16. (Previously Presented) The method of claim 1 wherein the remote key cannot be used to protect more than one memory region at a given instant.
17. (Currently Amended) The method of claim 1 further comprising:
reusing the remote key after the remote direct memory access operation from the second host processor endnode is completed.
18. (Currently Amended) The method of claim 1 further comprising:
disabling a translation for the remote key after the remote key is used for the remote direct memory access operation from the second host processor endnode.
19. (Currently Amended) A distributed computer system comprising:
communication fabric;
a first host processor endnode including:
a first processor;

Amendment and Response under 37 C.F.R. 1.116

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

a first memory configured to store a first consumer process which binds a remote key to a first address representing a contiguous memory address range accessible by the consumer process; and

a first interface controller sending the bound remote key and first address from the first host processor endnode on the communication fabric; and
a second host processor endnode including:

a second processor;

a second interface controller receiving the bound remote key and first address from the first host processor endnode via the communication fabric; and

a second memory configured to store a second consumer process performing a remote direct memory access operation from the second host processor endnode to access the contiguous memory address range including sending, via the second interface controller, the bound remote key and the first address from the second host processor endnode to the first host processor endnode on the communication fabric.

20. (Currently Amended) The distributed computer system of claim 19 wherein ~~first endnode further includes~~ the first memory is configured to store:

an operating system kernel process having a virtual memory manager controlling local memory access protection in the first host processor endnode.

21. (Currently Amended) The distributed computer system of claim 19 wherein in the remote direct memory operation if the bound remote key and corresponding first address supplied by the second host processor endnode do not match the bound remote key and first address- in the first host processor endnode, the second host processor endnode is not granted access to the contiguous memory address range.

22. (Currently Amended) The distributed computer system of claim 19 wherein in the remote direct memory operation if the contiguous memory address range represented by the first address bound to the remote key supplied by the second host processor endnode is invalid, the second host processor endnode is not granted access to the contiguous memory address range.

Amendment and Response under 37 C.F.R. 1.116

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

23. (Currently Amended) The distributed computer system of claim 19 wherein the first address is an effective address pointing to an address space in memory accessible by the first consumer process.

24. (Previously Presented) The distributed computer system of claim 23 wherein the effective address points to a virtual address space.

25. (Currently Amended) The distributed computer system of claim 19 wherein the first consumer process is a user process.

26. (Currently Amended) The distributed computer system of claim 19 wherein the first consumer process is a kernel process.

27. (Currently Amended) The distributed computer system of claim 26 wherein the first address is a virtual address accessible by the first consumer process which is a consumer kernel process.

28. (Currently Amended) The distributed computer system of claim 19 wherein at the consumer process employs a bind remote key verb to bind the first address to the remote key.

29. (Currently Amended) The distributed computer system of claim 19 wherein at the consumer process employs an allocate remote key verb to obtain at least one remote key.

30. (Currently Amended) The distributed computer system of claim 19 wherein at the consumer process employs an unbind remote key verb to unbind the remote key from the first address.

31. (Currently Amended) The distributed computer system of claim 29 wherein at the consumer process employs a deallocate remote key verb to retire at least one remote key that was previously obtained via the allocate remote key verb.

Amendment and Response under 37 C.F.R. 1.116

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

32. (Previously Presented) The distributed computer system of claim 19 wherein the remote key cannot be used to protect more than one memory region at a given instant.

33. (Currently Amended) The distributed computer system of claim 19 wherein the first consumer process reuses the remote key after the remote direct memory access operation from the second host processor endnode is completed.

34. (Currently Amended) The distributed computer system of claim 19 wherein the remote key is a one-shot validating mechanism such that a translation for the remote key is disabled after the remote key is used for the remote direct memory access operation from the second host processor endnode.